

# Clinical Reminders Index Technical Manual



**May 2021**

Provider Systems  
Office of Information and Technology  
Department of Veterans Affairs

## REVISION HISTORY

Date	Pages	Description	Developer/ Technical Writer
May 2021	<a href="#">17</a> , <a href="#">35</a>	Changes to 4.9. Added 5.3	REDACTED
Nov 2016	<a href="#">all</a>	Applied new OI&T format and remediated for Section 508	REDACTED EPMO/BAM
Aug 2016	<a href="#">14</a>	Added index for V Imm Contra/Refusal Events file (PX*1*215)	REDACTED
Feb 2016	<a href="#">14</a> , <a href="#">23</a> , <a href="#">26</a>	Change to V Immunization index, per PX*1.0*210	REDACTED
May 2014		Many updates per Developer review	REDACTED
Feb 2014	<a href="#">2</a>	Overview of updates per the Reminders ICD-10 Update project	REDACTED
Feb 2014	<a href="#">17</a>	Change to Problem List index, per GMPL*2.0*44	REDACTED
Feb 2014	<a href="#">17</a>	Change to Registration index, per DG*5.3*862	REDACTED
May 2012		Corrected codes in Registration Index text	REDACTED
Oct 2008	<a href="#">14</a>	Updated MH Index	REDACTED
Dec 2009	<a href="#">24</a>	Update MH Index in Summary Table	REDACTED
Dec 2004		Original release of Reminders Index (PXR*1*12)	REDACTED

## Table of Contents

<b>1. Clinical Reminders Index Global .....</b>	<b>1</b>
1.1. Introduction.....	1
1.2. Global Placement .....	1
1.3. Journaling.....	1
1.4. Changes to Reminder Indexes made by the Clinical Reminders ICD-10 Update project .....	2
1.4.1. Patches in the Clinical Reminders project build .....	2
1.4.2. DG*5.3*862.....	2
1.4.3. GMPL*2.0*44.....	2
1.5. Clinical Reminders Index Management.....	3
1.6. PXR INDEX BUILD Option .....	4
1.6.1. Error Messages .....	5
1.7. Inpatient Pharmacy.....	7
1.8. Outpatient Pharmacy.....	8
<b>2. Disable/Enable Reminder Evaluation .....</b>	<b>8</b>
2.1. Protocols .....	9
<b>3. PXR INDEX COUNT Option.....</b>	<b>12</b>
<b>4. Index Details .....</b>	<b>12</b>
4.1. Inpatient Pharmacy.....	12
4.2. Non-VA Meds .....	12
4.3. Outpatient Pharmacy.....	12
4.4. Order Entry.....	13
4.5. Lab .....	13
4.6. Mental Health .....	14
4.7. PCE .....	14
4.8. Problem List .....	17
4.9. Radiology .....	17
4.10.PTF.....	17
4.11.Vitals.....	23
<b>5. Cross-References .....</b>	<b>27</b>

5.1. Using FileMan to obtain detailed Cross-Reference descriptions .....	31
5.1.1. Method 1, Inquire Option .....	31
5.1.2. Method 2, Data Dictionary Utility .....	32
5.2. LAB Details.....	34
5.2.1. Routines.....	34
5.2.2. Lab Indexes .....	35
5.3. Order Entry Details .....	35

# 1. Clinical Reminders Index Global

## 1.1. Introduction

The Clinical Reminders Index global has been designed to provide an index of clinical data, which supports rapid access to clinical data. It is used by Clinical Reminders v2.0, which evaluates reminders in 1/3 to 1/2 the time that v1.5 required. A large part of the speedup can be attributed to the Index, because it provides such an efficient way to find patient data. The Index is a resource that can be used by other packages or site-developed applications whenever they need to find patient data. Use of the Index is supported by subscription to ICR #4290. This document describes the structure of the Index and how to use it, as well as how to populate it and manage it.

The basic structure of the index is:

```
^PXRMINDEX(FILE NUMBER,"IP",ITEM,DFN,DATE,DAS)
```

```
^PXRMINDEX(FILE NUMBER,"PI",DFN,ITEM,DATE,DAS)
```

Where “IP” stands for item and patient and “PI” for patient and item.

The “IP” index lets you find all patients with a particular item, while the “PI” index lets you find all items for a patient. DAS stands for DA string, similar to FileMan’s DA array. It is a semicolon-separated string that specifies the exact location in the source global where the data is stored. This can be as simple as the internal entry number (IEN) or can have a number of pieces (for example, a lab test result has four pieces).

Indexes with items that are coded values, such as ICD codes, vary from the basic structure, in order to support look-ups based on data such as primary diagnosis or principal procedure. Details are found in the sections describing each index.

## 1.2. Global Placement

This global serves as an index for the clinical data in a number of packages, so it is independent of a particular package. When new data is entered into the globals being indexed, the Index will grow, so it needs to be placed where it has room to grow.

There is a utility for estimating the initial size of the Index. To run this utility at the programmer prompt, type D ESTTASK^PXRMISE. This will start a TaskMan job that estimates the initial size of the index for each global as well as the total size. The information will be delivered in a MailMan message sent to members of the mail group defined in file #800. The estimated sizes will be given as a number of blocks which are 2K for Caché sites.

We recommend placing ^PXRMINDEX in its own volume set. The initial size of the dataset should be based on the estimated size, plus leaving room for growth.

## 1.3. Journaling

Journaling of PXRMINDEX is NOT required, because the index can be rebuilt, if necessary.

## 1.4. Changes to Reminder Indexes made by the Clinical Reminders ICD-10 Update project

To support the introduction of ICD-10 codes into VistA, Clinical Reminders has taken a very general approach, wherein Clinical Reminders taxonomies are being restructured to be Lexicon-based instead of pointer-based. This allows the use of any coding system supported by the Lexicon package. In addition to adding ICD-10 codes, SNOMED CT codes are being added. With the release of CPRS 29, SNOMED CT codes can be collected by Problem List and Clinical Reminders will be able to search for them.

### 1.4.1. Patches in the Clinical Reminders project build

Patch PX\*1.0\*211 adds a new V-file named V STANDARD CODES that can store codes from any coding system supported by the Lexicon.

The sources of coded data that are indexed are Problem List, PTF, V CPT, V POV and V STANDARD CODES.

For these files, with some exceptions, the first two subscripts of the Index will become

```
^PXRMINDEX(FILE NUMBER,CODING SYSTEM)
```

Where coding system is the standard three-character Source Abbreviation defined in file #757.03. The exceptions apply to V CPT and V POV and are described in the PCE section.

### 1.4.2. DG\*5.3\*862

This build updates the Clinical Reminders Index cross-references in the PTF file (#45) to accommodate ICD-10 diagnosis and procedure codes. It restructures the PTF portion of the Clinical Reminders Index to a generic format that can support all ICD coding systems. This format is:

```
^PXRMINDEX(45,CODING SYSTEM,"INP",CODE,NODE,DFN,DATE,DAS)
```

```
^PXRMINDEX(45,CODING SYSTEM,"PNI",DFN,NODE,CODE,DATE,DAS)
```

Where CODING SYSTEM is a three-character abbreviation as defined in the Coding Systems file (#757.03) and CODE is the code, not the pointer.

The post-install routine will start a background job to rebuild the file #45 index in the new format.

### 1.4.3. GMPL\*2.0\*44

This build updates the Clinical Reminders Index cross-references in the Problem file (#9000011) to accommodate ICD-10 CM diagnosis codes. It restructures the Problem List portion of the Clinical Reminders Index to a generic format that can support ICD and SNOMED CT coding systems. This format is:

```
^PXRMINDEX(9000011,CODING SYSTEM,"ISPP",CODE,STATUS,PRIORITY,DFN,DLM,DAS)
```

```
^PXRMINDEX(9000011,CODING SYSTEM,"PSPI",DFN,STATUS,PRIORITY,CODE,DLM,DAS)
```

Where CODING SYSTEM is a three-character abbreviation as defined in the Coding Systems file (#757.03) and CODE is the code, not the pointer. The post-install routine will start a background job to rebuild the file #9000011 index in the new format.

## 1.5. Clinical Reminders Index Management

The option, PXR INDEX MANAGEMENT, is a menu containing PXR INDEX BUILD and PXR INDEX COUNT.

The index building utility serves two purposes:

1. It initially populates the indexes by indexing the existing data. It works its way through the entire global, putting entries in the index for each piece of unique patient data it finds.
2. If the index ever gets corrupted or destroyed, the utility can be used to rebuild the index. Therefore, there is no need to journal the index, since it can be recreated from the original data at any time.

When the index utility finishes indexing a particular global, it sets the following three nodes:

```
^PXRINDEX(FILE NUMBER,"GLOBAL NAME")=$GET1^DID(FILE NUMBER,"","","GLOBAL
NAME")
^PXRINDEX(FILE NUMBER,"BUILT BY")=DUZ
^PXRINDEX(FILE NUMBER,"DATE BUILT")=$NOW^XLFTD
```

In addition to providing information about who built the index and when it got populated, these nodes can be used to determine when the index is complete and ready for use.

When data is added, edited, or deleted, the indexes are kept current using new-style MUMPS cross-references. These cross-references call APIs that set and kill the index entries.

## 1.6. PXR INDEX BUILD Option

The index build utility can be accessed through the option PXR INDEX MANAGEMENT or directly via PXR INDEX BUILD.

```
Select OPTION NAME: PXR INDEX BUILD          PXR INDEX BUILD          run routine
PXR INDEX BUILD
Which indexes do you want to (re)build?
  1 - LABORATORY TEST (CH, Anatomic Path, Micro)
  2 - MENTAL HEALTH
  3 - MENTAL HEALTH (MHA3)
  4 - ORDER
  5 - PTF
  6 - PHARMACY PATIENT
  7 - PRESCRIPTION
  8 - PROBLEM LIST
  9 - RADIOLOGY
 10 - V CPT
 11 - V EXAM
 12 - V HEALTH FACTORS
 13 - V IMMUNIZATION
 14 - V PATIENT ED
 15 - V POV
 16 - V SKIN TEST
 17 - V STANDARD CODES
 18 - VITAL MEASUREMENT
Enter your list: (1-18):
```

It can be used to build or rebuild the indexes for each of the globals, in any combination you want. You can run the build interactively or as a tasked job. Rebuilding an index is not something you would normally want to do. If problems are found with some entries in an index it is better to deal with them on an individual basis. Rebuilding an entire index is a last resort when there is no other way to repair or restore it.

After selecting the globals to be indexed, you will be given the choice of submitting a tasked job or running it interactively. In either case, when the index build completes, members of the mail group defined in file #800 will be sent a MailMan message that looks something like:

```
Subj: Index for global AUPNVPED successfully built [#12184]
10/03/17@10:25 5 lines
From: POSTMASTER (Sender: DOE,JOHN) In 'IN' basket. Page 1 *New*
-----
Build of Clinical Reminders index for global AUPNVPED completed.
Build finished at 10/03/2017@10:25:27
288 entries were created.
Elapsed time: 1 secs
0 errors were encountered.
```

**Note:** If the person who builds the indexes is not a member of the mail group, the messages will not be delivered to the members of the mail group, unless it is a public mail group.

For large globals, the build time can be many hours, so you may want to schedule these builds for overnight or weekends when the system is not under heavy use. The indexes for smaller globals, which require few block splits, can be built quickly (a few minutes), so you have wider latitude in when to build those.

### 1.6.1. Error Messages

If any entries couldn't be indexed, the completion message will look something like:

```
Subj: Index for global PS(55 successfully built [#12187] 10/03/17@10:30 6 lines

From: POSTMASTER (Sender: DOE,JOHN) In 'IN' basket. Page 1 *New*

-----

Build of Clinical Reminders index for global PS(55 completed.
Build finished at 10/03/2017@10:30:07
6416 entries were created.
Elapsed time: 13 secs
136 errors were encountered.
```

Another MailMan message contains the error information for up to the last N errors. The error information starts with the most recent entry in the global that has an error and progressively works back toward older entries. The number of errors displayed, N, is a site-configurable parameter. The parameter is stored in the CLINICAL REMINDERS PARAMETERS file, #800, in the field MAXIMUM NUMBER OF INDEX ERRORS, field #15 of file #800, Clinical Reminder Parameters. The default value is 200. If you wish to change this, you can use the ENTER OR EDIT FILE ENTRIES FileMan option. If you find a substantial number of errors, it is likely there is a systematic problem, so after determining the cause and solution for the first few errors, you can probably apply the same corrective action to the bulk of them.

The message has the format: global, entry identification, and error message. The error message describes the problem – for example, missing or invalid data. Here is a sample of errors you might see for file #55:

```
Subj: CLINICAL REMINDER INDEX BUILD ERROR(S) [#14895] 11/19/17@11:50 136 lines
```

```
From: POSTMASTER (Sender: DOE,JOHN) In 'IN' basket. Page 1
```

```
-----  
GLOBAL: PS(55 ENTRY: DFN=1 D1=33 IV missing stop date  
GLOBAL: PS(55 ENTRY: DFN=1 D1=34 IV missing stop date  
GLOBAL: PS(55 ENTRY: DFN=1 D1=35 IV missing stop date  
GLOBAL: PS(55 ENTRY: DFN=1 D1=36 IV missing stop date  
GLOBAL: PS(55 ENTRY: DFN=2 D1=1 IV missing stop date  
GLOBAL: PS(55 ENTRY: DFN=3 D1=1719 Unit Dose missing stop date  
GLOBAL: PS(55 ENTRY: DFN=3 D1=1 IV missing stop date  
GLOBAL: PS(55 ENTRY: DFN=3 D1=2 IV missing stop date  
GLOBAL: PS(55 ENTRY: DFN=3 D1=3 IV missing stop date  
GLOBAL: PS(55 ENTRY: DFN=4 D1=2 D2=1 IV additive missing drug  
GLOBAL: PS(55 ENTRY: DFN=4 D1=8 D2=1 IV additive missing drug  
GLOBAL: PS(55 ENTRY: DFN=6 D1=5 D2=1 IV additive missing drug  
GLOBAL: PS(55 ENTRY: DFN=6 D1=7 D2=1 IV additive missing drug  
GLOBAL: PS(55 ENTRY: DFN=6 D1=590 IV missing stop date  
GLOBAL: PS(55 ENTRY: DFN=8 D1=1 D2=1 IV additive missing drug  
GLOBAL: PS(55 ENTRY: DFN=9 D1=2 IV missing stop date  
GLOBAL: PS(55 ENTRY: DFN=9 D1=7 D2=1 IV additive missing drug  
GLOBAL: PS(55 ENTRY: DFN=10 D1=31 IV missing stop date  
GLOBAL: PS(55 ENTRY: DFN=10 D1=32 IV missing stop date
```

The entry information identifies the exact entry in the global that could not be indexed.

If we examine the last line, it tells us that for ^PS(55,10,"IV",32,0), there is no stop date, so it can't be indexed. If you are not familiar with the structure of the global being indexed, it will be helpful to have a data dictionary listing on hand to help you interpret the entry identification information.

Entries that are not indexable will never be found by any application that uses the index to find data. Each site should make a decision concerning what they want to do about non-indexable entries.

## 1.7. Inpatient Pharmacy

Sites have reported that the bulk of the errors are the same type: missing start date or missing patient.

```
GLOBAL: ^PS(55, ENTRY: DFN=994 D1=1545755 Unit Dose missing start date
Global ^PS(55,994,,1545755
      PS(55,994,,1545755
^PS(55,994,5,1545755,0) = ^^^^^^^^E
^PS(55,994,5,1545755,9,0) = ^55.09D^1^1

GLOBAL: ^PS(55, ENTRY: DFN=388 D1=1804771 Unit Dose missing start date
Global ^PS(55,388,,1804771
      PS(55,388,,1804771
^PS(55,388,5,1804771,0) = ^^^^^^^^E
^PS(55,388,5,1804771,9,0) = ^55.09D^1^1

GLOBAL: ^PS(55, ENTRY: DFN=1096 D1=1819001 Unit Dose missing start date
Global ^PS(55,1096,,1819001
      PS(55,1096,,1819001
^PS(55,1096,5,1819001,0) = ^^^^^^^^E
^PS(55,1096,5,1819001,9,0) = ^55.09D^1^1
Global ^

GLOBAL: ^PS(55, ENTRY: DFN=1245 D1=16 IV missing start date
Global ^PS(55,1245,,16
      PS(55,1245,,16
^PS(55,1245,"IV",16,0) = 16^^^P^^^INFUSE VIA MINIPUMP^NOW^^1000^^^0^^S
X=PSSTATUS
^PS(55,1245,"IV",16,1) = ONE TIME
^PS(55,1245,"IV",16,2) = 2861123.1008^1^IBG
^PS(55,1245,"IV",16,3) = PROTECT FROM LIGHT/DO NOT REFRIGERATE
^PS(55,1245,"IV",16,6) = 67^20 MG
^PS(55,1245,"IV",16,"A",0) = ^55.04A^1^1
^PS(55,1245,"IV",16,"A",1,0) = 1^D^IBG^COMPUTER DOWN^2861123.1045
^PS(55,1245,"IV",16,"A",2,1,0) = ^55.15^1^1
^PS(55,1245,"IV",16,"A",2,1,1,0) = STATUS^DISCONTINUED^
^PS(55,1245,"IV",16,"AD",0) = ^55.02PA^1^1
^PS(55,1245,"IV",16,"AD",1,0) = 108^20 MG
^PS(55,1245,"IV",16,"SOL",0) = ^55.11IPA^1^1
^PS(55,1245,"IV",16,"SOL",1,0) = 1^10 ML
```

## 1.8. Outpatient Pharmacy

Sites have reported that the bulk of the errors are the same type: missing drug or missing patient.

```
GLOBAL: ^PSRX( ENTRY: 200167 missing drug^PSRX(200167,0)="
^154^^^^^^^^^^^^^2881118^^^^^^1"
      2)="2881118^2881127^^^^^^1^ "
      3)=2881127
      "POE")=1
      "SIG")="^0"
      "STA")=11
      "TYPE")=0
```

```
GLOBAL: ^PSRX( ENTRY: 5287379 missing DFN
```

```
Global ^PSRX(5287379
      PSRX(5287379
^PSRX(5287379,0) = 5285540
^PSRX(5287379,2) = ^^^^^^^^^^
^PSRX(5287379,3) =
^PSRX(5287379,"POE") = 1
```

```
GLOBAL: ^PSRX( ENTRY: 5288355 missing DFN
```

```
Global ^PSRX(5288355
      PSRX(5288355
^PSRX(5288355,0) = 5285807
^PSRX(5288355,2) = ^^^^^^^^^^
^PSRX(5288355,3) =
^PSRX(5288355,"POE") = 1
```

## 2. Disable/Enable Reminder Evaluation

The option PXR INDEX BUILD provides the ability to rebuild selected portions of the Clinical Reminders Index. While an index is rebuilding, any reminder that uses the data from that index cannot be correctly evaluated – it will have the status of CNBD (cannot be determined). In the past, a MailMan message was sent to the Clinical Reminders mail group every time a reminder could not be evaluated because an index was rebuilding. Now, when an index is going to be rebuilt, reminder evaluation will be automatically disabled, meaning that any attempt to evaluate a reminder will result in an immediate return of a CNBD status. The Clinical Maintenance display will include text letting the user know that reminder evaluation is disabled and the reason(s). When the index has finished rebuilding, evaluation will be automatically enabled.

The option PXRМ DISABLE/ENABLE EVALUATION provides a manual disable/enable function. If for some reason, reminder evaluation needs to be disabled, it can be done through this option. This option should be given to a very limited number of people and can only be used by holders of the PXRМ MANAGER key. When the issue that required disabling evaluation has been handled, reminder evaluation can be enabled again using this same option. Note that this option can be used to enable evaluation even if it was not disabled using this option. For example, if reminder evaluation was automatically disabled for an index rebuild, this option could be used to enable evaluation even though the index is still rebuilding. If that is done, the MailMan messages will start being sent again.

When reminder evaluation is disabled, the following options and protocols will be put out of order.

Options:

PXRМ DEF INTEGRITY CHECK ALL
PXRМ DEF INTEGRITY CHECK ONE
PXRМ ORDER CHECK TESTER
PXRМ REMINDERS DUE
PXRМ REMINDERS DUE (USER)

## 2.1. Protocols

PXRМ PATIENT LIST CREATE

PXRМ EXTRACT MANUAL TRANSMISSION

When reminder evaluation is again enabled, these options and protocols will be put back in order.

Anytime reminder evaluation is disabled, a message with the subject “REMINDER EVALUATION DISABLED” will be sent to the Clinical Reminders mail group. The message will give the date and time that evaluation was disabled, list the reasons for disabling evaluation, and a search will be made for any Clinical Reminders TaskMan jobs that could be affected. There will be a list of those that are found; it will include the job description, the status (pending or running), and the task number. The results of any jobs that are already running will be unreliable and should be discarded. If possible, these jobs should be stopped, so that they don’t waste system resources. None of the pending jobs should be allowed to start until evaluation is enabled again.

When evaluation is enabled, a message with the subject “REMINDER EVALUATION ENABLED” will be sent to the Clinical Reminders mail group. It will contain the date and time evaluation was disabled and when it was enabled. This gives you the exact period of when evaluation was disabled.

Here are examples of disable and enable messages:

MailMan message for CRMANAGER,TWO

Printed at CPRS30.FO-SLC.MED.VA.GOV 04/16/17@10:32

Subj: REMINDER EVALUATION DISABLED [#122941] 04/16/14@10:30 58 lines

From: POSTMASTER (Sender: CRMANAGER, ONE) In 'IN' basket. Page 1

-----  
Reminder evaluation was disabled on Apr 16, 2017@10:30:42.

Because of this, the following TaskMan jobs can produce erroneous results.

Pending jobs should not be allowed to start until evaluation is enabled.

The results of running jobs should be discarded and if possible, running jobs should be stopped.

Reason: index rebuild for file #45.

Reminders Due Report Jobs

Task number - 316820

Status - Active: Running

Time - Feb 08, 2012@12:40:28

User - CRCOORDINATOR, TWO

Reminder Patient List Jobs

Task number - 1980022

Status - Active: Running

Time - Apr 16, 2014@08:00

User - TASKMAN,PROXY USER

Task number - 1980207

Status - Active: Pending

Time - Apr 17, 2014@08:00

User - TASKMAN,PROXY USER

Reminder Extract Jobs

Task number - 342256

Status - Active: Pending

Time - Mar 06, 2012@20:04:25

User - CRCOORDINATOR, SIX

Task number - 1867565

Status - Active: Pending

Time - May 17, 2013@16:44:13

User - CRCOORDINATOR, SIX

Task number - 1902474

Status - Active: Pending

Time - Jul 17, 2013@17:16:30

User - CRCOORDINATOR, TEN

Task number - 1945932

Status - Active: Pending

Time - Oct 22, 2013@12:37:23

User - CRCOORDINATOR, THIRTY

Task number - 1946204

Status - Active: Pending

Time - Oct 23, 2013@12:37:35

User - CRCOORDINATOR, TWO

Task number - 1966964

Status - Active: Pending

Time - Feb 05, 2014@07:54:39

User - CRCOORDINATOR, THREE

Enter message action (in IN basket): Ignore//

Subj: REMINDER EVALUATION ENABLED [#122942] 04/16/17@10:30 2 lines

From: POSTMASTER (Sender: CRMANAGER, ONE) In 'IN' basket. Page 1

-----  
Reminder evaluation was enabled on Apr 16, 2017@10:30:49.

It was disabled on Apr 16, 2017@10:30:42.

Enter message action (in IN basket): Ignore//

### 3. PXR INDEX COUNT Option

The index count utility can be accessed through the option PXR INDEX MANAGEMENT or directly via PXR INDEX COUNT. This utility provides a count by year of the entries in the index. This will let sites see how their data is distributed on a yearly basis.

The selection for the count utility is identical to the build utility and the results are sent in a MailMan message just like the results of the build utility.

## 4. Index Details

### 4.1. Inpatient Pharmacy

The index is on file 55, Pharmacy Patient. The structure is:

```
^PXRINDEX(55,"IP",DRUG,DFN,START,STOP,DAS)
^PXRINDEX(55,"PI",DFN,DRUG,START,STOP,DAS)
```

Where DRUG is a pointer to the Drug file (#50), START is the start date, and STOP is the stop date. The API to retrieve the associated data is OEL^PSJPXR1(DAS,.DATA).

It is documented in ICR #3836.

### 4.2. Non-VA Meds

The index is on the non-VA med multiple of file #55, Pharmacy Patient. The structure is:

```
^PXRINDEX("55NVA","IP",POI,DFN,START,STOP,DAS)
^PXRINDEX("55NVA","PI",DFN,POI,START,STOP,DAS)
```

Where POI is a pointer to the Pharmacy Orderable Item file. START is the start date, if it exists; otherwise it is the documented date. STOP is the discontinued date if it exists; otherwise is "U"\_D0. If STOP has the form "U"\_D0 it should be interpreted to mean that no discontinued date exists so the patient is currently taking the non-VA med.

The API to retrieve the associated data is NVA^PSOPXR1(DAS,.DATA). It is documented in ICR 3793.

### 4.3. Outpatient Pharmacy

The index is on file #52, Prescription file. The structure is:

```
^PXRINDEX(52,"IP",DRUG,DFN,START,STOP,DAS)
^PXRINDEX(52,"PI",DFN,DRUG,START,STOP,DAS)
```

Where DRUG is a pointer to Drug file, START is the starting date (RELEASE DATE) and STOP is the stop date ( RELEASE DATE + DAYS SUPPLY). The API to retrieve the associated data is PSRX^PSOPXR1(DAS,.DATA).

It is documented in ICR #3793.

## 4.4. Order Entry

The index is on file 100, Order file. The structure is:

```
^PXRMINDEX(100,"IP",OI,DFN,START,STOP,DAS)
^PXRMINDEX(100,"PI",DFN,OI,START,STOP,DAS)
```

Where OI is a pointer to the Orderable Items file, START is the START DATE, and STOP is the STOP DATE. The API to retrieve the associated data is EN^ORX8(DA). Note that DA is the first piece of DAS and the data is returned in the array ORUPCHK.

The API is documented in ICR #871.

## 4.5. Lab

The index is on file 63, Lab Data. The structure is:

Chemistry, Hematology, other Lab Tests

```
^PXRMINDEX(63,"IP",ITEM,DFN,DATE,DAS)
^PXRMINDEX(63,"PI",DFN,ITEM,DATE,DAS)
```

Microbiology and Anatomic Path data have an additional index

```
^PXRMINDEX(63,"PDI",DFN,DATE,ITEM,DAS)
```

Where DATE is the Date/Time of collection. The structure of ITEM depends on the type of lab data.

For Chemistry, Hematology, and other lab tests, ITEM is numeric and a pointer to the Laboratory Test file.

For Microbiology, ITEM is of the format

```
"M;[S T O A M];#".
```

where the middle section can be one of:

S is specimen (# is a pointer to the Topography [SNOMED] file)

T is test (# is a pointer to the Laboratory Test file)

O is organism (# is a pointer to the Etiology Field [SNOMED] file)

A is antibiotic (# is a pointer to the Antimicrobial Susceptibility file)

M is a TB drug (# is the field number of the TB drug - ^DD(63.39,).

For Anatomic Pathology, ITEM is of the format

```
"A;[S T O D M E F P I];#".
```

where the middle section can be one of:

S is specimen (# is 1.free text value)

T is test (# is a pointer to the Laboratory Test file)

O is organ/tissue (# is a pointer to the Topography [SNOMED] file)

D is disease (# is a pointer to Disease Field [SNOMED] file)

M is morphology (# is a pointer to the Morphology Field [SNOMED] file)  
 E is etiology (# is a pointer to the Etiology Field [SNOMED] file)  
 F is function (# is a pointer to the Function [SNOMED] file)  
 P is procedure (# is a pointer to the Procedure [SNOMED] file)  
 I is ICD (# is a pointer to the ICD DIAGNOSIS file)

Microbiology and Anatomic Pathology data are stored in a complex hierarchy. The ITEM is therefore a compound expression. This allows single elements of data to be easily found. The DAS also depends on the type of lab data. A chemistry test result has four semicolon pieces. Microbiology and Anatomic Pathology can be more complex and have a much more nested structure.

The API to retrieve the associated data is LRPXRM^LRPXAPI(.DATA,DAS,ITEM). This information should be reviewed in the context of other data associated with the specimen. The API is documented in ICR #4245. The Lab package has other APIs that use these indexes.

## 4.6. Mental Health

The index is on file 601.84, MH Administrations. The structure is:

```
^PXRMINDEX(601.84,"IP",INS,DFN,DATE,DAS)
^PXRMINDEX(601.84,"PI",DFN,INS,DATE,DAS)
```

Where INS is a pointer to the MH Instrument file #601. The API to retrieve the associated data is DENDAS71^YTQPXRM6(.DATA,DAS).

The API is documented in ICR #5043.

## 4.7. PCE

There are indexes on all of the V files, with the exception of V Provider and V Treatment. There are two types of indexes for the V files – one for coded values and one for the non-coded values. There is a third type of index for the V Imm Contra/Refusal Events file.

The non-coded values are stored in the following V files:

V FILE	FILE NUMBER
V EXAM	9000010.13
V HEALTH FACTORS	9000010.23
V IMMUNIZATION	9000010.11
V PATIENT ED	9000010.16
V SKIN TEST	9000010.12

The structure of the index for the non-coded values is:

```

^PXRMINDEX(FILE NUMBER,"IP",ITEM,DFN,DATE,DAS)
^PXRMINDEX(FILE NUMBER,"PI",DFN,ITEM,DATE,DAS)

```

Where item is the .01 of the V file (for example, a pointer to the Education Topic file or Health Factor file), and DATE is the Event Date and Time, if it is populated, otherwise it is the Visit/Admit Date&Time.

The V Immunization file has an additional index:

```

^PXRMINDEX(9000010.11,"CVX","IP",CVX_CODE,DFN,DATE,DAS)
^PXRMINDEX(9000010.11,"CVX","PI",DFN,CVX_CODE,DATE,DAS)

```

CVX is the Center for Disease Control’s vaccine administered code.

Note: The “ACR” cross-reference on the Immunization file updates the ^PXRMINDEX(9000010.11,"CVX") index when an immunization’s CVX code is changed.

Coded values are stored in the following V files:

V FILE	FILE NUMBER	CODE TYPE
V CPT	9000010.18	CPT-4
V POV	9000010.07	ICD diagnosis
V STANDARD CODES	9000010.71	Any standard code

Because of the large number of entries, the existing structure of the Index for ICD-9 diagnosis and CPT-4 codes will be left as is:

```

^PXRMINDEX(FILE NUMBER,"IPP",CODEP,TYPE,DFN,DATE,DAS)
^PXRMINDEX(FILE NUMBER,"PPI",DFN,TYPE,CODEP,DATE,DAS)

```

Where CODEP is a pointer to the coded value, TYPE is primary procedure for V CPT and primary/secondary for V POV. DATE is the Visit Date.

Starting with the ICD-10 update the structure of the Index will become:

```

^PXRMINDEX(FILE NUMBER,CODING SYSTEM,"IPP"CODE,TYPE,DFN,DATE,DAS)
^PXRMINDEX(FILE NUMBER,CODING SYSTEM,"PPI",DFN,TYPE,CODE,DATE,DAS)

```

Where CODING SYSTEM is the three-character Source Abbreviation from file #757.03.

For example, the structure of the V POV Index for ICD-10 diagnosis codes is:

```

^PXRMINDEX(9000010.07,"10D","IPP",CODE,TYPE,DFN,DATE,DAS)
^PXRMINDEX(9000010.07,"10D","PPI",DFN,TYPE,CODE,DATE,DAS)

```

Where CODE is the ICD-10 the code.

The V Imm Contra/Refusal Events is stored in the following V file:

<b>V FILE</b>	<b>FILE NUMBER</b>
<b>V IMM CONTRA/REFUSAL EVENTS</b>	9000010.707

The structure of the index is:

PXRMINDX(9000010.707, "PIC", DFN, IMM, CONTRA/REFUSAL, START, STOP, DAS)
PXRMINDX(9000010.707, "PCI", DFN, CONTRA/REFUSAL, IMM, START, STOP, DAS)
PXRMINDX(9000010.707, "ICP", IMM, CONTRA/REFUSAL, DFN, START, STOP, DAS)
PXRMINDX(9000010.707, "CIP", CONTRA/REFUSAL, IMM, DFN, START, STOP, DAS)

Where IMM is a pointer to the Immunization file; Contra/Refusal is a variable pointer to the Imm Contraindication Reasons (#920.4) or Imm Refusal Reasons (#920.5) files; START is the Event Date and Time, or if Event Date and Time is not populated, the Visit Date/Time will be used instead; STOP is the Warn Until Date, or if Warn Until Date is not populated, 9999999 will be used instead.

The APIs for retrieving the associated date are in routine PXPXRM. There is an entry point for each V file; these are listed in the following table. The APIs are documented in ICR #4250.

<b>V FILE</b>	<b>PXPXRM ENTRY POINT</b>
<b>V CPT</b>	VCPT(DAS,.DATA)
<b>V EXAM</b>	VXAM(DAS,.DATA)
<b>V HEALTH FACTORS</b>	VHF(DAS,.DATA)
<b>V IMMUNIZATION</b>	VIMM(DAS,.DATA)
<b>V IMM CONTRA/REFUSAL EVENTS</b>	VICR(DAS,.DATA)
<b>V PATIENT ED</b>	VPEDU(DAS,.DATA)
<b>V POV</b>	VPOV(DAS,.DATA)
<b>V STANDARD CODES</b>	VSCDATA(DAS,.DATA)
<b>V SKIN TEST</b>	VSKIN(DAS,.DATA)

## 4.8. Problem List

The structure of the index is:

```
^PXRMINDEX(9000011,CODING SYSTEM,"ISPP",CODE,STATUS,PRIORITY,DFN,DLM,DAS)
^PXRMINDEX(9000011,CODING SYSTEM,"PSPI",DFN,STATUS,PRIORITY,CODE,DLM,DAS)
```

Where CODING SYSTEM is a three-character abbreviation as defined in the Coding Systems file (#757.03) and CODE is the code, not the pointer. STATUS is the status of the problem, either active ("A") or inactive ("I"). PRIORITY can be acute ("A"), chronic ("C"), or null, in which case a "U" is stored. DLM is the Date Last Modified. This structure lets you quickly do things like find active problems whose status is acute.

The API to retrieve the associated data is PROBDATA^GMPLPXR, it is documented in ICR #5881.

## 4.9. Radiology

The index is on file 70, Rad/Nuc Med Patient. The structure is:

```
^PXRMINDEX(70,"IP",PROC,DFN,DATE,DAS)
^PXRMINDEX(70,"PI",DFN,PROC,DATE,DAS)
```

Where PROC is a pointer to the Rad/Nuc Med Procedures file. The API to retrieve the associated data is EN1^RAPXR(DAS,.DATA).

The API is documented in ICR #3731.

## 4.10. PTF

The structure of the index is:

```
^PXRMINDEX(45,CODING SYSTEM,"INP",CODE,NAME,DFN,DATE,DAS)
^PXRMINDEX(45,CODING SYSTEM,"PNI",DFN,NAME,CODE,DATE,DAS)
```

Where CODING SYSTEM is a three-character abbreviation as defined in the Coding Systems file (#757.03) and CODE is the code, not the pointer. NAME is the name of the field where the code is stored.

ICD codes are stored in a number of fields in PTF so the storage node information (NODE) is included in the Index to allow quick searching and retrieval of specific nodes. The following tables summarize the fields that are indexed and their node names.

<b>Field Number</b>	<b>Field Name</b>	<b>Name</b>
<b>79</b>	PRINICIPAL DIAGNOSIS	DXLS
<b>80</b>	PRINICIPAL DIAGNOSIS pre 1986	PDX
<b>79.16</b>	SECONDARY DIAGNOSIS 1	D SD1
<b>79.17</b>	SECONDARY DIAGNOSIS 2	D SD2
<b>79.18</b>	SECONDARY DIAGNOSIS 3	D SD3
<b>79.19</b>	SECONDARY DIAGNOSIS 4	D SD4
<b>79.201</b>	SECONDARY DIAGNOSIS 5	D SD5
<b>79.21</b>	SECONDARY DIAGNOSIS 6	D SD6
<b>79.22</b>	SECONDARY DIAGNOSIS 7	D SD7
<b>79.23</b>	SECONDARY DIAGNOSIS 8	D SD8
<b>79.24</b>	SECONDARY DIAGNOSIS 9	D SD9
<b>79.241</b>	SECONDARY DIAGNOSIS 10	D SD10
<b>79.242</b>	SECONDARY DIAGNOSIS 11	D SD11
<b>79.243</b>	SECONDARY DIAGNOSIS 12	D SD12
<b>79.244</b>	SECONDARY DIAGNOSIS 13	D SD13
<b>79.245</b>	SECONDARY DIAGNOSIS 14	D SD14
<b>79.246</b>	SECONDARY DIAGNOSIS 15	D SD15
<b>79.247</b>	SECONDARY DIAGNOSIS 16	D SD16

<b>79.248</b>	SECONDARY DIAGNOSIS 17	D SD17
<b>79.249</b>	SECONDARY DIAGNOSIS 18	D SD18
<b>79.2491</b>	SECONDARY DIAGNOSIS 19	D SD19
<b>79.24911</b>	SECONDARY DIAGNOSIS 20	D SD20
<b>79.24912</b>	SECONDARY DIAGNOSIS 21	D SD21
<b>79.24913</b>	SECONDARY DIAGNOSIS 22	D SD22
<b>79.24914</b>	SECONDARY DIAGNOSIS 23	D SD23
<b>79.24915</b>	SECONDARY DIAGNOSIS 24	D SD24
<b>45.02,5</b>	ICD 1	M ICD1
<b>45.02,6</b>	ICD 2	M ICD2
<b>45.02,7</b>	ICD 3	M ICD3
<b>45.02,8</b>	ICD 4	M ICD4
<b>45.02,9</b>	ICD 5	M ICD5
<b>45.02,11</b>	ICD 6	M ICD6
<b>45.02,12</b>	ICD 7	M ICD7
<b>45.02,13</b>	ICD 8	M ICD8
<b>45.02,14</b>	ICD 9	M ICD9
<b>45.02,15</b>	ICD 10	M ICD10
<b>45.02,81.01</b>	ICD 11	M ICD11
<b>45.02,81.02</b>	ICD 12	M ICD12
<b>45.02,81.03</b>	ICD 13	M ICD13

<b>45.02,81.04</b>	ICD 14	M ICD14
<b>45.02,81.05</b>	ICD 15	M ICD15
<b>45.02,81.06</b>	ICD 16	M ICD16
<b>45.02,81.07</b>	ICD 17	M ICD17
<b>45.02,81.08</b>	ICD 18	M ICD18
<b>45.02,81.09</b>	ICD 19	M ICD19
<b>45.02,81.1</b>	ICD 20	M ICD20
<b>45.02,81.11</b>	ICD 21	M ICD21
<b>45.02,81.2</b>	ICD 22	M ICD22
<b>45.02,81.3</b>	ICD 23	M ICD23
<b>45.02,81.4</b>	ICD 24	M ICD24
<b>45.02,81.4</b>	ICD 25	M ICD25

## ICD Operation/Procedure Codes

<b>Field Number</b>	<b>Field Name</b>	<b>Node Name</b>
45.05,4	PROCEDURE CODE 1	P1
45.05,5	PROCEDURE CODE 2	P2
45.05,6	PROCEDURE CODE 3	P3
45.05,7	PROCEDURE CODE 4	P4
45.05,8	PROCEDURE CODE 5	P5
45.05,9	PROCEDURE CODE 6	P6
45.05,10	PROCEDURE CODE 7	P7
45.05,11	PROCEDURE CODE 8	P8
45.05,12	PROCEDURE CODE 9	P9
45.05,13	PROCEDURE CODE 10	P10
45.05,14	PROCEDURE CODE 11	P11
45.05,15	PROCEDURE CODE 12	P12
45.05,16	PROCEDURE CODE 13	P13
45.05,17	PROCEDURE CODE 14	P14
45.05,18	PROCEDURE CODE 15	P15
45.05,19	PROCEDURE CODE 16	P16
45.05,20	PROCEDURE CODE 17	P17
45.05,21	PROCEDURE CODE 18	P18
45.05,22	PROCEDURE CODE 19	P19
45.05,23	PROCEDURE CODE 20	P20
45.05,24	PROCEDURE CODE 21	P21
45.05,25	PROCEDURE CODE 22	P22

<b>45.05,26</b>	PROCEDURE CODE 23	P23
<b>45.05,27</b>	PROCEDURE CODE 24	P24
<b>45.05,28</b>	PROCEDURE CODE 25	P25
<b>45.01,8</b>	OPERATION CODE 1	S1
<b>45.01,9</b>	OPERATION CODE 2	S2
<b>45.01,10</b>	OPERATION CODE 3	S3
<b>45.01,11</b>	OPERATION CODE 4	S4
<b>45.01,12</b>	OPERATION CODE 5	S5
<b>45.01,13</b>	OPERATION CODE 6	S6
<b>45.01,14</b>	OPERATION CODE 7	S7
<b>45.01,15</b>	OPERATION CODE 8	S8
<b>45.01,16</b>	OPERATION CODE 9	S9
<b>45.01,17</b>	OPERATION CODE 10	S10
<b>45.01,18</b>	OPERATION CODE 11	S11
<b>45.01,19</b>	OPERATION CODE 12	S12
<b>45.01,20</b>	OPERATION CODE 13	S13
<b>45.01,21</b>	OPERATION CODE 14	S14
<b>45.01,22</b>	OPERATION CODE 15	S15
<b>45.01,23</b>	OPERATION CODE 16	S16
<b>45.01,24</b>	OPERATION CODE 17	S17
<b>45.01,25</b>	OPERATION CODE 18	S18
<b>45.01,26</b>	OPERATION CODE 19	S19
<b>45.01,27</b>	OPERATION CODE 20	S20
<b>45.01,28</b>	OPERATION CODE 21	S21

<b>45.01,29</b>	OPERATION CODE 22	S22
<b>45.01,30</b>	OPERATION CODE 23	S23
<b>45.01,31</b>	OPERATION CODE 24	S24
<b>45.01,32</b>	OPERATION CODE 25	S25

In summary, there are 25 ICD diagnosis codes that can be associated with the discharge. The date for the discharge entries is the discharge date. Movement data is stored in subfile #45.02 so there are 25 possible ICD diagnosis codes for each movement. The date is the MOVEMENT DATE field #45.02, 10.

The names DXLS and PDX don't bear any resemblance to the field names because these fields were renamed from DXLS to PRINCIPAL DIAGNOSIS and PDX to PRINCIPAL DIAGNOSIS pre 1986 after the Index was released.

The ICD procedure codes stored in subfiles #45.05 and #45.01 are indexed. The dates for these are the PROCEDURE DATE field #45.05.01 and SURGERY/PROCEDURE DATE field #45.01.01 respectively.

The ICD procedure codes stored in the 401P node (fields #45.01, #45.02, #45.03, #45.04, and #45.050 are not indexed because they are valid only for admissions prior to 10/01/1987. This is documented in the User Manual – PTF Menu; located in the Admission Discharge Transfer (ADT) section of the VistA Documentation Library.

The ICD diagnosis codes stored in the CPT RECORD DATE/TIME multiple, field #45.06,.04 are not indexed because CPT data is stored in file #46 not file #45.

The API to retrieve the data is PTF^DGPXRM(DAS,.DATA), it is documented in ICR #4457.

## 4.11. Vitals

The index is on file 120.5, GMRV Vital Measurement. The structure is:

```
^PXRMINDEX(120.5,"IP",VITAL TYPE,DFN,DATE,DAS)
^PXRMINDEX(120.5,"PI",DFN,VITAL TYPE,DATE,DAS)
```

Where VITAL TYPE is a pointer to the GMRV Vital Type file #120.51. Entries that are marked as "entered-in-error" are not indexed. The API to retrieve the associated data is EN^GMRVPXRM(GMRVDATA,DAS).

The API is documented in ICR #3647.

### Summary of the detailed index structure given above

Package	Structure	Pointer	API	ICR #
<b>Inpatient Pharmacy</b>	^PXRMINDEX(55,"IP",DRUG,DFN,START,STOP,DAS)  ^PXRMINDEX(55,"PI",DFN,DRUG,START,STOP,DAS)	DRUG points to Drug file		3836
<b>Lab</b>	^PXRMINDEX(63,"IP",ITEM,DFN,DATE,DAS)  ^PXRMINDEX(63,"PI",DFN,ITEM,DATE,DAS)  ^PXRMINDEX(63,"PDI",DFN,,DATE,ITEM,DAS)	ITEM is formatted depending on the type of data	LRPXRML RPXAPI(.DATA,DAS,ITEM)	4245
<b>Mental Health</b>	^PXRMINDEX(601.84,"IP",INS,DFN,DATE,DAS)  ^PXRMINDEX(601.84,"PI",DFN,INS,DATE,DAS)  ^PXRMINDEX(601.2,"IP",INS,DFN,DATE,DAS)  ^PXRMINDEX(601.2,"PI",DFN,INS,DATE,DAS)	INS pointer to the MH Instrument file	ENDAS71^ YTQPXRM 6(.DATA,DAS)	5043
<b>Non-VA meds</b>	^PXRMINDEX("55NVA","IP",POI,DFN,START,STOP,DAS)  ^PXRMINDEX("55NVA","PI",DFN,POI,START,STOP,DAS)		NVA^PSO PXRM1(DAS,.DATA)	3793
<b>Order Entry</b>	^PXRMINDEX(100,"IP",OI,DFN,DATE,DAS)  ^PXRMINDEX(100,"PI",DFN,OI,DATE,DAS)	OI points to the Orderable Items file	EN^ORX8(DAS)	871
<b>Outpatient Pharmacy</b>	^PXRMINDEX(52,"IP",DRUG,DFN,START,STOP,DAS)  ^PXRMINDEX(52,"PI",DFN,DRUG,START,STOP,DAS)	DRUG is a pointer to Drug file	PSRX^PS OPXRM1(DAS,.DATA)	3793

<p><b>PCE</b></p> <p><b>V FILES:</b></p> <p><b>V CPT</b></p> <p><b>V EXAM</b></p> <p><b>V HEALTH FACTORS</b></p> <p><b>V IMMUNIZATION</b></p> <p><b>V PATIENT ED</b></p> <p><b>V POV</b></p> <p><b>V SKIN TEST V IMM CONTRA/REFUSAL EVENTS</b></p> <p><b>V STANDARD CODES</b></p>	<p>Non-coded values:</p> <p>^PXRMINDEX(FILE NUMBER,"IP",ITEM,DFN,DATE,DAS)</p> <p>^PXRMINDEX(FILE NUMBER,"PI",DFN,ITEM,DATE,DAS)</p> <p>Coded values:</p> <p>^PXRMINDEX(FILE NUMBER,CODING SYSTEM,"IPP",CODE,TYPE,DFN,DATE,DAS)</p> <p>^PXRMINDEX(FILE NUMBER,CODING SYSTEM,"PPI",DFN,TYPE,CODE,DATE,DAS)</p> <p>V Immunization:</p> <p>^PXRMINDEX(9000010.11,"CVX","IP",CVX_CODE,DFN,DATE,DAS)</p> <p>^PXRMINDEX(9000010.11,"CVX","PI",DFN,CVX_CODE,DATE,DAS)</p> <p>V Imm Contra/Refusal Events:</p> <p>^PXRMINDEX(9000010.707,"PIC",DFN,IMM,CONTRA/REFUSAL,START,STOP,DAS)</p> <p>^PXRMINDEX(9000010.707,"PCI",DFN,CONTRA/REFUSAL,IMM,START,STOP,DAS)</p> <p>^PXRMINDEX(9000010.707,"ICP",IMM,CONTRA/REFUSAL,DFN,START,STOP,DAS)</p> <p>^PXRMINDEX(9000010.707,"CIP",CONTRA/REFUSAL,IMM,DFN,START,STOP,DAS)</p>	<p>Item is the .01 of the V file, for example a pointer to the Education Topic file or Health Factor file</p>	<p>PXPXRM ENTRY POINT</p> <p>VCPT(DAS,DATA)</p> <p>VXAM(DAS,DATA)</p> <p>VHF(DAS,DATA)</p> <p>VIMM(DAS,DATA)</p> <p>VPEDU(DAS,DATA)</p> <p>VPOV(DAS,DATA)</p> <p>VSCDATA(DAS,DATA)</p> <p>VSKIN(DAS,DATA)</p> <p>VICR(DAS,DATA)</p>	<p>4250</p>
<p><b>Problem List</b></p>	<p>^PXRMINDEX(9000011,CODING SYSTEM,"ISPP",CODE,STATUS,PRIORITY,DFN,DLM,DAS)</p> <p>^PXRMINDEX(9000011,CODING SYSTEM,"PSPI",DFN,STATUS,PRIORITY,CODE,DLM,DAS)</p>	<p>None</p>	<p>PROB^GM PLPXR</p>	<p>5881</p>

<b>Radiology</b>	^PXRMINDEX(70,"IP",PROC,DFN,DATE,DAS) ^PXRMINDEX(70,"PI",DFN,PROC,DATE,DAS)	PROC points to the Rad/Nuc Med Procedures file	EN1^RAPXRM(DAS,.DATA)	3731
<b>Registration</b>	^PXRMINDEX(45,CODING SYSTEM,"INP",CODE,DFN,DATE,NODE,DAS) ^PXRMINDEX(45,CODING SYSTEM,"PNI",DFN,CODE,DATE,NODE,DAS) NODE,DAS)	None	PTF^DGPXRM(DAS,.DATA)	
<b>Vitals</b>	^PXRMINDEX(120.5,"IP",VITAL TYPE,DFN,DATE,DAS) ^PXRMINDEX(120.5,"PI",DFN,VITAL TYPE,DATE,DAS)	VITAL TYPE points to the GMRV Vital Type file	EN^GMRVPXR(.GM RVDATA,DAS)	3647

## 5. Cross-References

The Index is kept current by using new-style FileMan cross-references that fire whenever data is added, edited, or deleted. A list of the cross-references follows.

**NOTE:** Some of the packages do direct sets of the data into their globals instead of using FileMan. In those cases, the routines where the data is set or killed have been modified to call the package API that does the set or kill of the Index entry.

File #	File Name	Sub-file #	Sub-file Name	Cross-reference
45	PTF			ACRDSD1 ACRDSD2 ACRDSD3 ACRDSD4 ACRDSD5 ACRDSD6 ACRDSD7 ACRDSD8 ACRDSD9 ACRDSD10 ACRDSD11 ACRDSD12 ACRDSD13 ACRDSD14 ACRDSD15 ACRDSD16 ACRDSD17 ACRDSD18 ACRDSD19 ACRDSD20 ACRDSD21 ACRDSD22 ACRDSD23 ACRDSD24 ACRDDXLS ACRDPDX

		45.01	401	ACRPS1 ACRPS2 ACRPS3 ACRPS4 ACRPS5 ACRPS6 ACRPS7 ACRPS8 ACRPS9 ACRPS10 ACRPS11 ACRPS12 ACRPS13 ACRPS14 ACRPS15 ACRPS16 ACRPS17 ACRPS18 ACRPS19 ACRPS20 ACRPS21 ACRPS22 ACRPS23 ACRPS24 ACRPS25
		45.02	501	ACRDM1 ACRDM2 ACRDM3 ACRDM4 ACRDM5 ACRDM6 ACRDM7 ACRDM8 ACRDM9 ACRDM10 ACRDM11 ACRDM12 ACRDM13

				ACRDM14 ACRDM15 ACRDM16 ACRDM17 ACRDM18 ACRDM19 ACRDM20 ACRDM21 ACRDM22 ACRDM23 ACRDM24 ACRDM25
		45.05	601	ACRPP1 ACRPP2 ACRPP3 ACRPP4 ACRPP5 ACRPP6 ACRPP7 ACRPP8 ACRPP9 ACRPP10 ACRPP11 ACRPP12 ACRPP13 ACRPP14 ACRPP15 ACRPP16 ACRPP17 ACRPP18 ACRPP19 ACRPP20 ACRPP21 ACRPP22 ACRPP23 ACRPP24 ACRPP25
<b>52</b>	Prescription			ACRO

		52.1	Refill	ACRR
		52.2	Partial Date	ACRP
<b>55</b>	Pharmacy Patient	55.01	IV	ACRIV
		55.05	Non-VA meds	ACRNVA
		55.06	Unit Dose	ACRUD
<b>63</b>	Lab Data			None
<b>70</b>	Rad/Nuc Med Patient	70.03	Examinations	ACR
<b>100</b>	Order			None
<b>120.5</b>	GMRV Vital Measurement			ACR
<b>601.2</b>	Psych Instrument Patient	601.22	Date	ACR
<b>9000010.07</b>	V POV			ACR
<b>9000010.11</b>	V IMMUNIZATION			ACR
<b>9000010.12</b>	V SKIN TEST			ACR
<b>9000010.13</b>	V EXAM			ACR
<b>9000010.16</b>	V PATIENT ED			ACR
<b>9000010.18</b>	V CPT			ACR
<b>9000010.23</b>	V HEALTH FACTORS			ACR
<b>9000010.707</b>	V IMM CONTRA/REFUSAL EVENTS			ACR
<b>9000010.71</b>	V STANDARD CODES			ACR
<b>9000011</b>	Problem			ACR01
<b>9999999.14</b>	Immunization			ACR

## 5.1. Using FileMan to obtain detailed Cross-Reference descriptions

If you wish a more detailed description of any of these cross-references, there are two different ways to get it – both use FileMan.

### 5.1.1. Method 1, Inquire Option

Use the Inquire option on the Index file.

```
VA FileMan 22.0

Select OPTION: I   INQUIRE TO FILE ENTRIES
OUTPUT FROM WHAT FILE: INDEX//
Select INDEX: ACR
   1  ACR   120.5  Clinical Reminders cross-reference.
   2  ACR   70    Clinical Reminders index.
   3  ACR  9000010.18  Clinical Reminders index.
   4  ACR  9000010.23  Clinical Reminders index.
   5  ACR  9000010.11  Clinical Reminders index.
Press <RETURN> to see more, '^' to exit this list, OR
CHOOSE 1-5:
   6  ACR  9000010.16  Clinical Reminders index.
   7  ACR  9000010.07  Clinical Reminders index.
   8  ACR  9000010.12  Clinical Reminders index.
   9  ACR  9000010.13  Clinical Reminders index.
  10  ACR   601.2    Clinical Reminders cross-reference.
Press <RETURN> to see more, '^' to exit this list, OR
CHOOSE 1-10: 6  9000010.16  ACR  Clinical Reminders index.
ANOTHER ONE:
STANDARD CAPTIONED OUTPUT? Yes//   (Yes)
Include COMPUTED fields: (N/Y/R/B): NO// BOTH Computed Fields and Record
Number
(IEN)

NUMBER: 279                                FILE: 9000010.16
NAME: ACR
SHORT DESCRIPTION: Clinical Reminders index.
TYPE: MUMPS                                EXECUTION: RECORD
ACTIVITY: IR                               ROOT TYPE: INDEX FILE
```

```

ROOT FILE: 9000010.16                USE: ACTION
DESCRIPTION:  This cross-reference builds two indexes. One for finding all
patients with a particular education topic and one for finding all the
education topics a patient has.  The indexes are stored in the Clinical
Reminders index global as:
  ^PXRMINDEX(9000010.16,"IP",EDUCATION TOPIC,DFN,VISIT DATE,DAS) and
  ^PXRMINDEX(9000010.16,"PI",DFN,EDUCATION TOPIC,VISIT DATE,DAS) respectively.
SET LOGIC: D SVFILE^PXPXRM(9000010.16,.X,.DA)
KILL LOGIC: D KVFILE^PXPXRM(9000010.16,.X,.DA)
KILL ENTIRE INDEX CODE: K ^PXRMINDEX(9000010.16)
ORDER NUMBER: 1                        TYPE OF VALUE: FIELD
  FILE: 9000010.16                      FIELD: .01
  SUBSCRIPT NUMBER: 1                   COLLATION: forwards
ORDER NUMBER: 2                        TYPE OF VALUE: FIELD
  FILE: 9000010.16                      FIELD: .02
  SUBSCRIPT NUMBER: 2                   COLLATION: forwards
ORDER NUMBER: 3                        TYPE OF VALUE: FIELD

```

### 5.1.2. Method 2, Data Dictionary Utility

The other way to obtain detailed cross-reference descriptions is to use the Data Dictionary Utility. You can look at an entire file or a sub-file. In the example below, we look at a sub-file.

```

VA FileMan 22.0

Select OPTION:  DATA DICTIONARY UTILITIES
Select DATA DICTIONARY UTILITY OPTION:  LIST FILE ATTRIBUTES
  START WITH WHAT FILE: PTF//
    GO TO WHAT FILE: PTF//
    Select SUB-FILE: 601
Select LISTING FORMAT: STANDARD// INDEXES ONLY
What type of cross-reference (Traditional or New)? Both// NEW
Which field: ALL//
DEVICE:  ANYWHERE    Right Margin: 80//
NEW-STYLE INDEX LIST -- FILE #45                                08/18/04    PAGE 1
-----
Subfile #45.05

```

New-Style Indexes:

ACRPP1 (#1287)    RECORD    MUMPS    IR    ACTION    WHOLE FILE (#45)

Short Descr:    Clinical Reminders Index for ICD procedure code lookup.

Description:    This cross-reference builds two indexes, one for finding all patients with a particular ICD procedure code and one for finding all the ICD procedure codes a patient has. The indexes are stored in the Clinical Reminders Index global as:

^PXRMINDEX(45,CODESYS,"INP",CODE,NODE,DFN,DATE,DAS) and

^PXRMINDEX(45,CODESYS,"PNI",DFN,NODE,CODE,DATE,DAS)

respectively. CODESYS is the standard three-character abbreviation for the coding system. DATE is the surgery/procedure date. NODE is P (for procedure) followed by procedure code number. For example, P1 means it was found on the P node and it was Procedure Code 1. For complete details, see the Clinical Reminders Index Technical Guide/Programmer's Manual.

Set Logic:    D SPTFP^DGPTDDCR(.X,.DA,"P",1)

Kill Logic:    D KPTFP^DGPTDDCR(.X,.DA,"P",1)

    X(1):    PROCEDURE DATE    (45.05,.01)    (Subscr 1)    (forwards)

    X(2):    PROCEDURE CODE 1    (45.05,4)    (Subscr 2)    (forwards)

The rest of the cross-references on this sub-file have been left out for brevity.

If you know the field number or field name of a field used in the cross-reference, you can select a single cross-reference for display.

VA FileMan 22.0

Select OPTION:    DATA DICTIONARY UTILITIES

Select DATA DICTIONARY UTILITY OPTION:    LIST FILE ATTRIBUTES

  START WITH WHAT FILE: PTF//

    GO TO WHAT FILE: PTF//

  Select SUB-FILE:

Select LISTING FORMAT: STANDARD// INDEXES ONLY

What type of cross-reference (Traditional or New)? Both// NEW

Which field: ALL//    SECONDARY DIAGNOSIS 1

DEVICE:    ANYWHERE    Right Margin: 80//

NEW-STYLE INDEX LIST -- FILE #45, FIELD #79.16                    08/18/04    PAGE 1

ACRSDS1 (#1263)	RECORD	MUMPS	IR	ACTION
Short Descr:	Clinical Reminders Index for ICD diagnosis code lookup.			
Description:	<p>This cross-reference builds two indexes, one for finding all patients with a particular ICD diagnosis code and one for finding all the ICD diagnosis codes a patient has. The indexes are stored in the Clinical Reminders Index global as:</p> <p><code>^PXRMINDEX(45,CODESYS,"INP",CODE,NAME,DFN,DATE,DAS)</code> and  <code>^PXRMINDEX(45,CODESYS,"PNI",DFN,NAME,CODE,DATE,DAS)</code></p> <p>respectively. CODESYS is the standard three character abbreviation for the coding system. DATE is the discharge date. If it does not exist then the admission date is used.</p> <p>NAME is the name of the field where the code is stored. An example is D SD1, where D SD signifies it is a discharge secondary diagnosis. If the TYPE OF RECORD is CENSUS then the entry is not indexed. For complete details, see the Clinical Reminders Index Technical Guide/Programmer's Manual.</p>			
Set Logic:	D SPTFDD^DGPTDDCR(.X,.DA,"D SD1")			
Kill Logic:	D KPTFDD^DGPTDDCR(.X,.DA,"D SD1")			
	X(1):	PATIENT (45,.01)	(Subscr 1)	(forwards)
	X(2):	ADMISSION DATE (45,2)	(Subscr 2)	(forwards)
	X(3):	TYPE OF RECORD (45,11)	(Subscr 3)	(forwards)
	X(4):	SECONDARY DIAGNOSIS 1 (45,79.16)	(Subscr 4)	(forwards)
	X(5):	DISCHARGE DATE (45,70)		

## 5.2. LAB Details

Lab results are stored in the Lab Data file #63. This is a very hierarchical file with a strong dependence on the data dictionary.

The Lab package makes programming calls to update the ^PXRMINDEX indexes. Chemistry-type data updates the indexes when results are verified. Anatomic Pathology and Microbiology update indexes when results are reported and/or released. Any changes to existing lab data update the indexes. All indexes are set using SLAB^LRPX and killed using KLAB^LRPX.

### 5.2.1. Routines

Chemistry-type data updates in a central routine.

**LRVER3A** Chemistry data are updated on verification and editing of verified data. All transactions go through LRVER3A, which stores the results and sets all the cross-references. This routine calls CHSET^LRPX.

**LROC** It is very rare but chemistry data may be purged during purging of old orders and accessions. This only happens on data that is corrupted and not reportable. This routine calls CHKILL^LRPX.

All Microbiology and Anatomic Pathology data are updated using the same routine, UPDATE^LRPXR. Adding, editing, or deleting data invokes this call. Results are compared before and after editing. Any change will update the indexes. This routine is called from several routines:

- LRAPDA
- LRAPDSR
- LRMIEDZ
- LRMIEDZ2
- LRMISTF1
- LRMIV
- LRMIV1
- LRMIV2

### 5.2.2. Lab Indexes

<code>^PXRMINDEX(63, "PI", DFN, ITEM, DATE, DAS)</code>
---

This index is used for finding results of tests on a patient.

<code>^PXRMINDEX(63, "IP", ITEM, DFN, DATE, DAS)</code>
---

This index is used for finding patients that have specific lab results.

<code>^PXRMINDEX(63, "PDI", DFN, DATE, ITEM, DAS)</code>
--

This index is only used for Microbiology and Anatomic Pathology and is used for finding results on a patient for a specific time period. Chemistry-type data does not require this because the data are already stored in a similar format in the Lab Data file. Micro and AP data use a compound structure for ITEM (the lab test or other coded result) and the "PDI" index provides a faster path to the results. Also, AP data is broken into four sections: Autopsy, Cytology, Electron Microscopy, and Surgical Pathology. This index collates results by collection date/time regardless of the section; again, making retrieval faster.

### 5.3. Order Entry Details

Many of the complex indexes used in Order Entry were created before new-style cross-references existed, therefore the cross-references that set and kill them had to be implemented as MUMPS cross-references. When the routines that set and kill the "AE" and "AR" indexes are called, all the data needed to set and kill the OR portion of the Clinical Reminders Index is available so it was natural to include the sets and kills of the Clinical Reminders Index.

AE	MUMPS
	Field: STOP DATE (100,22)
	Description: ^OR(100,"AE",ORSTOP,ORIFN) Allows retrieval of orders by expiration date; set only for orders that have not already completed, expired, or been discontinued or canceled.
	1)= D ES^ORDD100A
	2)= D EK^ORDD100A

The “AE” cross-reference calls ES^ORDD100A (set) and EK^ORDD100A (kill) and these call SOR^ORPXRM and KOR^ORPXRM to perform the sets and kills of the Clinical Reminders Index.

AR	MUMPS
	Field: OBJECT OF ORDER (100,.02)
	Description: ^OR(100,"AR",ORVP,9999999-ORRDT,ORIFN,ORDA) Allows retrieval of orders by inverse date released.
	1)= N ORDA S ORDA=0 F S ORDA=\$O(^OR(100,DA,8,ORDA)) Q:ORDA '>0 D RS^ORDD100(DA,ORDA,X)
	2)= N ORDA S ORDA=0 F S ORDA=\$O(^OR(100,DA,8,ORDA)) Q:ORDA '>0 D RK^ORDD100(DA,ORDA,X)

The “AR” cross-reference calls RS^ORDD100 (set) and RK^ORDD100 (kill). RS^ORDD100 calls PXRMAADD^ORDD100 which then calls SOR^ORPXRM. RK^ORDD100 calls PXRMKILL^ORDD100 which then calls KOR^ORPXRM.